User-defined Literals Solutions

Copyright © James Raynard 2021

Suffixes

- Write a program which creates the following variables and prints out their values
  - Float with value 3.14159
  - std::string with value "Hello"
  - Unsigned long long with value 12345678901234567890

Copyright © James Raynard 2021

User-defined literals

- What is meant by a user-defined literal?
  - A user-defined literal is an operator
  - The operator is called when a literal is followed by a suffix with the same name
- Why are user-defined literals useful?
  - They can be used to give meaning to "bare numbers", e.g. to represent physical units
  - They can be used to perform conversions
  - User-defined literals can be combined with classes to prevent ambiguous "bare numbers" being passed through interfaces (Mars Rover situations)

Copyright © James Raynard 2021

Floating-point literal example

- Write a literal operator which can be used as a suffix to a floating-point number representing a temperature in Fahrenheit
- The result should be a floating-point number representing the equivalent type in Celsius
- Write a program to test your literal operator

Copyright © James Raynard 2021

User-defined literals and type safety

- Why is the following interface potentially dangerous?

  ```
  void func(double temperature);
  ```

  - There is nothing to prevent a caller using the wrong units, e.g. using 32 as the temperature of frozen water instead of 0 (assuming that func expects an argument in Celsius)
- Write a suitable class and user-defined literal to remove the danger. Explain how these can be used to make the interface safe
  - Create a "Celsius" class and a user-defined literal which returns an instance of the class
  - Change the interface to take a Celsius object instead of double
  - Now, the function can only be called with a Celsius object, or a Celsius literal

Copyright © James Raynard 2021

String literal example

- Write a literal operator which can be used as a suffix for a C-style string
- The result should be an std::string object which contains the characters in reverse order
- Write a program to test your literal operator

Copyright © James Raynard 2021

"Raw" literals

- What is meant by a "raw" literal?
  - If we want a literal which does not correspond to the four types provided by C++, we use a raw literal
  - This allows us to create literals for our own classes
  - The argument to the literal operator is a const char *. This contains the characters in the literal, without any processing
  - Usually, the literal operator returns an object of the class, initialized using the value of the literal

Copyright © James Raynard 2021

"Raw" literals

- Write a class whose constructor takes a const char * and prints out its argument
- Write a literal operator for this class
- Write a program to test your literal operator

Copyright © James Raynard 2021